AMENDMENTS TO THE CLAIMS:

1-6. (Cancelled).

7. (New) A method for reading microprocessor's instructions, comprising the steps of:

executing a current instruction, pre-fetching and pre-decoding a next instruction following the current instruction in a current instruction cycle; and

setting an instruction reading-amount register to a first state if the next instruction pre-decoded is a conditional branch instruction, and otherwise setting the instruction reading-amount register to a second state, said conditional branch instruction having a branched target in one of the two succeeding instructions after said conditional branch instruction;

wherein the two succeeding instructions after the next instruction are pre-fetched in a next instruction cycle if the instruction reading-amount register is in a first state, and otherwise one instruction is pre-fetched in the next instruction cycle.

8. (New) The method for reading microprocessor's instructions as claimed in claim 7, wherein a program counter contains an address value, and the address value is increased by 1 for pre-fetching an instruction in the next instruction cycle if the next instruction is pre-decoded to be a general logic instruction, a new address value contained in the next instruction is loaded to the program counter if the next instruction is pre-decoded to be an unconditional branch instruction, a new address value contained in the next instruction is loaded to the program counter if the next instruction is pre-decoded to be a CALL instruction, and a new address is popped

from a stack and loaded to the program counter if the next instruction is pre-decoded to be a RETURN instruction.

9. (New) The method for reading microprocessor's instructions as claimed in claim 7, wherein a binary value 1 or 0 is set in the instruction reading-amount register to represent the first or second state for pre-fetching two instructions or one instruction in the next instruction cycle.

10. (New) An architecture for reading microprocessor's instructions, comprising:

a plurality of address lines having an address value;

an incremental circuit for incrementing said address value;

a first multiplexer controlled by a least significant bit of the incremented address value for selecting either the address value of the address lines or the incremented address value to output an odd address;

a second multiplexer controlled by a least significant bit of the address value of the address lines for selecting either the address value of the address lines or the incremented address value to output an even address;

an odd address buffer register for buffering the odd address from the first multiplexer;

an even address buffer register for buffering the even address from the second multiplexer;

an odd-page memory portion addressed by the odd address buffered in the odd address buffer register for outputting an instruction stored in the odd-page memory portion;

an even-page memory portion addressed by the even address buffered in the even address buffer register for outputting an instruction stored in the even-page memory portion;

a third multiplexer for selecting the instruction from either the odd-page memory portion or the even-page memory portion;

an instruction buffer register for buffering the instruction selected by the third multiplexer;

a fourth multiplexer for enabling the even address buffer register to access the even-page memory portion based on an inverted value of the least significant bit of the address lines;

a fifth multiplexer for enabling the odd address buffer register to access the odd-page memory portion based on the least significant bit of the address lines;

a sixth multiplexer for controlling the third multiplexer based on the least significant bit of the address lines to select the instruction from the odd-page memory portion or the even-page memory portion;

an instruction reading-amount register for indicating an amount of instructions to be pre-fetched; and

a processing unit for executing arithmetic logic operations, controlling, and pre-fetching one or two instructions from the instruction buffer register;

wherein the processing unit pre-fetches and pre-decodes a next instruction following a current instruction in a current instruction cycle and sets the instruction

reading-amount register to a state for pre-fetching one or two instructions in a next instruction cycle.

11. (New) The architecture for reading microprocessor's instructions as claimed in claim 10, wherein the instruction reading-amount register is set to a state for pre-fetching two instructions in the next instruction cycle if the next instruction pre-decoded is a conditional branch instruction.

12. (New) The architecture for reading microprocessor's instructions as claimed in claim 10, wherein the address lines are controlled by a program counter, and the address value of the address lines is increased by 1 for pre-fetching an instruction in the next instruction cycle if the next instruction is pre-decoded to be a general logic instruction, a new address value contained in the next instruction is loaded to the program counter if the next instruction is pre-decoded to be an unconditional branch instruction, a new address value contained in the next instruction is loaded to the program counter if the next instruction is pre-decoded to be a CALL instruction, and a new address is popped from a stack and loaded to the program counter if the next instruction is pre-decoded to be a RETURN instruction.

13. (New) The method for reading microprocessor's instructions as claimed in claim 10, wherein a binary value 1 or 0 is set in the instruction reading-amount register to represent a first or second state for pre-fetching two instructions or one instruction in the next instruction cycle.